AMENDMENTS IN THE CLAIMS

Please amend the claims as follows:

---

1. (currently amended) A data processing system comprising:

an interconnect;

an in-order processor that ~~processes~~ issues all memory access requests in program order, wherein said processor issues said memory access requests from an instruction sequence only in said program order and accepts data retrieved by a first and a second memory access request into its execution units only in said program order;

a memory system coupled to said processor which supports completing memory access requests in a weakly consistent order; and

a controller associated with said processor that ~~issues~~ forwards said memory access requests to said memory system and which automatically places a barrier operation on said interconnect ~~in response to~~ following each issuance of a memory access request to said memory system ~~issued~~, wherein said barrier operation indicates a need to complete the data operations associated with the memory access requests in program order from the perspective of the processor.

2. (original) The data processing system of Claim 1, wherein said controller includes means for creating said barrier operations.

3. (currently amended) The data processing system of Claim 1, ~~wherein said controller includes~~ further comprising:

means for ignoring a pending barrier operation when a subsequent load request appears in the instruction sequence; and

means for speculatively issuing the subsequent load requests to said memory system ~~while a~~ before the pending barrier operation is ~~pending~~ completed, wherein said subsequent load request is speculative because said subsequent load request is issued before a previous memory access request that may invalidate or change data retrieved by said load request completes within the memory system.

4. (original) The data processing system of claim 3, wherein said controller includes means for allowing data returned by a speculatively issued load request to be utilized by said processor only when an acknowledgment is received from all barrier operations pending when said load was issued.

5. (currently amended) An in-order processor comprising:

an instruction sequencing unit (ISU) that receives memory access instructions in program order;

a load store unit (LSU) including a controller that issues memory access requests associated with said memory access instructions to an interconnect that couples said processor to a memory system and wherein said controller automatically places a barrier operation on said interconnect in response to each issuance of a memory access request, wherein all said memory access requests are forwarded to memory in the program order ~~issued~~.

6. (original) The processor of Claim 5, wherein said controller includes means for creating said barrier operations.

7. (currently amended) The ~~data processing system~~ processor of Claim 5, wherein said controller includes:

means for ignoring a pending barrier operation when a subsequent load request appears in the instruction sequence; and

means for speculatively issuing the subsequent load requests to said interconnect ~~while a~~ before the pending barrier operation is ~~pending~~ completed, wherein said subsequent load request is speculative because said subsequent load request is issued before a previous memory access request that may invalidate or change data retrieved by said load request completes within the memory system.

8. (original) The data processing system of claim 7, wherein said controller includes means for allowing data returned by a speculatively issued load request to be utilized by said processor only when an acknowledgment is received from all barrier operations pending when said load was issued.

9. (currently amended) A method of processing instructions in a data processing system having a memory system, said method comprising the steps of:

receiving an instruction sequence at a processor in program order, said instruction sequence including at least a first and a second memory access instruction;

in response to receipt of said memory access instruction, ~~creating~~ generating a memory access request and a barrier operation;

automatically ~~placing~~ initiating said barrier operation ~~on an interconnect~~ after said memory access request is issued to a memory system; and

upon completion of said barrier operation, completing said first and said second memory access request in program order at said processor.

10. (currently amended) The method of Claim 9, wherein said second memory access request is a load request and said method further ~~including the step of~~ comprises:

ignoring a pending barrier operation of said first memory access request when a subsequent memory access request in the instruction sequence is a load request; and

speculatively issuing said load request to said memory system ~~while a~~ before the pending barrier operation is ~~pending~~ completed, wherein said load request is speculative because said load request is issued before a previous memory access request that may invalidate or change data retrieved by said load request completes within the memory system.

11. (original) The method of Claim 10, further including the step of forwarding data returned by said speculatively issued load request to a register or execution unit of said processor, when an acknowledgment is received for said barrier operation.